# Workshop 08+09+10

# Parallel Image/Video Processing and Computer Arithmetic

# Parallel Image/Video Processing and Computer Arithmetic

Larry Davis[1], Jean-Marc Delosme[2], Francky Catthoor[3]

[1] University of Maryland, College Park, MA 20742, USA
[2] Yale University, New Haven, CT 06520, USA
[3] IMEC, B-3001 Leuven, Belgium

**Abstract.** This workshop covers a wide range of issues in software and hardware design, structured under three headings: parallel image processing, computer arithmetic and design automation for multimedia processors.

## 1 Parallel image processing

Image processing was one of the earliest driving applications for parallel computing. Originally, the massive computations associated with processing large scale images and, more recently, the addition of interest in very large databases of images (medical, satellite) have had a significant impact on the design and development of parallel systems. About half of the papers in this workshop directly address aspects of high performance computing systems for image processing.

"A high performance image database system for remotely sensed imagery" by Shock et al. addresses the problems associated with maintaining a database of so-called level 1B remotely sensed data, and accessing it and integrating it on demand to form more conventional level 2 image products. This system provides environmental scientists with the capability of posing both regional and global queries to the image database system, and to create image products that do not suffer from the data degradation problems associated with standard image products in which the sensed data has undergone a sequence of resampling and interpolation steps.

The majority of the papers deal with parallelizing fundamental algorithms in image and signal processing. Two papers address problems in intermediate level vision. Chung et al. describe parallel algorithms for grouping problems that arise in intermediate level vision. These algorithms are part of a system developed at the University of Southern California for the interpretation of remotely sensed images, and their high performance implementation is significant for efficient analysis of very large high resolution images. Guil and Zapata describe a pipelined algorithm for computing the Hough transform. This topic has received considerable attention in the image processing literature because the Hough transform is a fundamental tool for the recognition of parameterized structures in image.

Synthetic Aperture Radar is an important military sensor because of its ability to see in all weather and night conditions, and to be employed at large distances from targets. Problems related to the formation and processing of Synthetic Aperture Radar images are discussed by Apiani et al.

There has been much interest in fast image coding within the last few years for applications such as high resolution video teleconferencing. Image coding algorithms based on linear prediction and iterated functions are described by Distasi et al. Uhl discusses parallel algorithms using non-stationary MRA's for coding.

The last two papers present hardware and software architectures for parallel image processing. Progress on the massively parallel processor CAM is described in the paper by Ikenaga and Ogura. CAM is a state-of-the-art SIMD array, suitable for incorporation in embedded image processing systems. Both the architecture and some performance results are included in the paper. Finally, the paper by Crookes et al. discusses issues related to efficient compiling for image co-processors.

## 2 Computer arithmetic in context

Since it regroups many communities sharing common concerns, Euro-Par has a strong potential to foster a number of contributions with a significant interdisciplinary flavor. A step in this direction has been taken this year by encouraging, through the merging of workshops, exchanges about problems and solution techniques between researchers in parallel image/video processing and computer arithmetic. Clearly there is also some synergy between computer arithmetic and other areas represented at Euro-Par, such as instruction level parallelism (see [12]) and parallel numerical algorithms (see [2]), and the emphasis might shift toward these areas in future Euro-Par conferences. Yet, taken as a whole, this year's contributions appear closer to the image processing area and this introduction will provide a few pointers to illustrate that connection.

A large share of the research in computer arithmetic deals with the use of redundant number systems, which enable additions without carry propagation. Three papers in this volume present work along this line. While one develops algorithms for the addition of real and complex numbers represented in various bases [7], the other two elaborate algorithms based on sequences of additions and shifts for the computation of rotations [1] and of exponentials and logarithms [10]. The avoidance of carry propagation enables additions to be performed either in parallel with small delays, independent of wordlength [1], or *on-line*, i.e., serially from most to least significant digit [10]. Frougny shows in [7] the tight formal connection between these two modes of computation. Arrays of on-line adders enable the computation of Hadamard and Walsh transforms or simple filtering operations such as image smoothing by means of neighborhood averaging (mean filtering) or convolution by Roberts Cross/Prewitt/Sobel kernels (for gra-

dient computation) in such a way that higher level operations, also performed on-line, may be initiated soon after the MSDs of the transformed or filtered image are available. On-line adders whose base is a quadratic complex number [7] may be used, in concert with on-line multipliers for complex numbers, to perform in parallel operations in the frequency domain (filtering, spectral estimation). On-line complex multipliers may be constructed for the bases used in [7] and their performance characteristics should be compared with those of the multipliers presented in [11].

CORDICs (COordinate Rotation DIgital Computers) are specialized arithmetic processors which can perform either one of the following operations or both: *rotation*, performing the rotation of a two-dimensional vector by a given angle, and *vectoring*, transforming cartesian into polar coordinates. CORDICs for each operation have been used in computer graphics: rotation in [16] and vectoring in [14]. Operation time is shorter if one is just interested in computing Euclidean distances or vector norms as in [14] since, when executing a vectoring operation, after about half of the iterations needed to achieve a given accuracy on the angle this accuracy is already attained on the length. The paper by Antelo et al. [1] focuses on the rotation operation, showing how to reduce the number of iterations while retaining a short iteration time. Such processors could be used to perform pipelined or parallel FFTs on images. Some of the ideas in the paper might still be employed if one further wanted to exploit the a priori knowledge of the values of the rotation angles in the FFTs [5]. The extension of the work of Antelo et al. to the vectoring operation, and thus to the computation of distances, would be of great interest to basic image processing. Indeed distance computation is employed both in the frequency domain, in the typical case where radially symmetric filters are employed, and in the spatial domain, for instance when computing the Euclidean distance transform [4] or the gradient at each point (for image sharpening). The vectoring operation may also be used in parallel on the points of interest in an image to construct the Hough transform for straight line detection; the rotation operation may then be used for de-Houghing, i.e. mapping back into cartesian space. Moreover, when computing the Hotelling transform, the combination of vectoring and rotation operations enables efficient parallel implementations of Jacobi's eigen-decomposition method [15].

The algorithms for exponential and logarithm computation presented in [10] are similar to the CORDIC rotation and vectoring algorithms. However Nielsen and Muller's emphasis is on digit-serial, MSD first, computation rather than fast parallel computation. Arrays of such on-line computing units could be used in the homomorphic filtering approach to image enhancement, where the logarithm enables separate operation on illumination and reflectance components and the exponential effects recombination of the filtered components. By cascading stages built out of on-line operators both latency and communications are reduced. A stage computing exponentials in parallel could be employed to perform a gamma correction, i.e. correct for the non-linear response of photographic film, and be followed by other stages of on-line operators operating on the corrected image. A stage computing logarithms in parallel could also be employed to display image spectra or Fourier transforms evaluated by stages of on-line operators.

The last three papers on computer arithmetic are fairly representative of the concerns in that area other than those stemming from the use of redundant representations. The short paper by King and Swatzlander investigates by means of simulations the errors on a parallel multiplier output induced by a couple of simple truncation schemes employed to reduce the multiplier area. Work along this line is particularly useful for the design and analysis of implementations of filtering algorithms.

All the above algorithms assumed fixed-point representations, although they can be fairly easily extended to handle floating-point representations (see e.g. [9], [6]). The excellent paper by Obermann and Flynn [12] derives by stepwise transformations a couple of pipelined floating-point adder designs for high clock-rate, dynamic instruction scheduling, processors that achieve reduced average latency while preserving single-cycle, i.e. maximum, throughput. This contribution, which improves the performance of modern microprocessors with little additional hardware, should have a broad impact.

For classes of problems where the dynamic range of some variables is so large that the overflow and underflow limitations of floating-point systems become a serious nuisance, another representation, known as symmetric level-index (SLI) arithmetic, has been introduced by Clenshaw, Turner and colleagues. Three-dimensional semiconductor process simulation, for the modeling of fabrication processes, and 2-D or 3-D device simulation are among the applications which could benefit from the simplifications brought to software development by the complete avoidance of overflow/underflow problems afforded by the use of the SLI representation. The paper by Anuta et al. [2] provides a very good introduction to SLI computer arithmetic and proposes improved algorithms for addition/subtraction and multiplication/division using the SLI representation. These algorithms are costly but they, and also new algorithms for the dot product and saxpy operation, are shown to be well suited to parallel implementation.

## 3 Major directions in design automation for multi-media processors

A combination of several key features is present in embedded multi-media systems that makes them quite distinct from other systems:

1. the use of large multi-dimensional array-type data structures.
2. the need to satisfy hard real-time constraints, causing worst-case behavior to be more important than average behavior.
3. typically a high throughput in some of the modules, requiring for real-time implementation the exploitation of a certain amount of parallelism but not massive parallelism.
4. a data-flow which mixes manifest and data-dependent relations, so that compile-time analysis is difficult but feasible (and crucial) to perform.

5. usually the presence of a very diverse mixture of fixed-point data types, making strong data typing essential to obtain bit-true behavior.
6. a cost function to be optimized that combines area and power consumption, both of which depend on architecture parameters in a highly non-linear way.

These properties have led to the introduction of a number of programmable multi-media oriented processors and also of many customized VLSI processors. All of them exhibit a mix of weakly parallel functional units, typically heterogeneous, and a distributed memory and bus organization. Programmable examples of this target class are the C80 of TI [3] and the TriMedia of Philips [8] but several other processors have been announced recently [8]. In the customized target class, MPEG-type video compression processors are very popular [13] but many other video, image and audio processing architectures continue to be proposed.

The design or programming of such processors is very demanding and the design time becomes a true bottleneck. In order to deal with this, there is a clear need for more design support. In this workshop, several approaches are highlighted that address issues vital for solving this design automation problem.

Danckaert et al. deal with an extended parallel compilation methodology to incorporate the power- and area-related cost of signal storage and transfers in multi-media applications. Application studies in the areas of image and video processing systems indicate that 50 to 80% of these costs are due to memory and communication. To reduce this dominant cost, the authors propose to address the system-level storage organization for the multi-dimensional signals as a first step in the overall methodology to map these applications, even before parallelization and load balancing. In addition, they demonstrate the usefulness of this novel approach based on two realistic test-vehicles, namely a cavity detector for medical image processing and a quad-tree based image coding application.

Verhaegh et al. discuss multi-dimensional periodic scheduling for real-time signal processing applications operating on signal streams. The cost function to be minimized is the area, including both memory and arithmetic units. They show that this optimization problem is NP-hard, even under relaxed conditions. Several special cases are however solvable in polynomial time. In addition, heuristics are presented to solve the general case. The key feature is the combination of real-time constraints on periodic streams with the accounting of the costs of both storage and arithmetic. Heterogeneous processor architectures can be dealt with in combination with irregular algorithms processing on streams.

De Coster et al. present the issues related to bit-true modeling when strong data-typing is present. They emphasize (parallel) simulation on programmable processors. Currently, either these issues are neglected in compilers or they are treated in a totally local way. Here, the authors discuss a new global approach where the entire flow graph is analyzed. A key feature consists in the globally steered transformations on the flow-graph utilizing the flexibility present in the implementation of the data type casting on programmable operators like shifters. This approach leads to a significant speed-up of the simulation compared to the local approach.

## References


1. Antelo, E., Bruguera, J., Lang, T., Villalba, J., Zapata, E.: High radix CORDIC rotation based on selection by rounding. This volume.
2. Anuta, M., Lozier, D., Schabanel, N., Turner, P.: Basic linear algebra operations in SLI arithmetic. This volume.
3. Balmer, K., Ing-Simmons, N., Moyse, P., Robertson, I., Keay, J., Hammes, M., Oakland, E., Simpson, R., Barr, G., Roskell, D.: A single chip multi-media video processor. Proc. IEEE Custom Integrated Circuits Conf. (1994) 91–94
4. Chen, L., Chuang, H.: Designing systolic architectures for complete Euclidean distance transform. J. of VLSI Signal Processing. **10** (1995) 163–179
5. Despain, A.: Fourier transform computers using CORDIC iterations. IEEE Trans. on Computers. **C-23** (1974) 993–1001
6. Duprat, J., Fiallos-Aguilar, M., Muller, J.-M., Yeh, H.: Delays of on-line floating point operators in borrow-save representation. IFIP Workshop on Algorithms and Parallel VLSI Architectures. (1991)
7. Frougny, C.: Parallel and on-line addition in negative base and some complex number systems. This volume.
8. Halfhill, T., Montgomery, J.: Chip fashion: multi-media chips. Byte Magazine. (Nov. 1995) 171–178
9. Metafas, D., Goutis, C.: A floating-point advanced Cordic processor. J. of VLSI Signal Processing, **10** (1995) 53–65
10. Nielsen, A., Muller, J.-M.: On-line algorithms for computing exponentials and logarithms. This volume.
11. Nielsen, A., Muller, J.-M.: On-line operators for complex arithmetic. 2nd Int. Conf. on Massively Parallel Computing Systems. (1996)
12. Oberman, S., Flynn, M.: A variable latency pipelined floating-point adder. This volume.
13. Pirsch, P., Demassieux, N., Gehrke, W.: VLSI architectures for video compression - a survey. Proc. of the IEEE. **83**, **2** (1995) 220–246
14. Turkowski, K.: Anti-aliasing through the use of coordinate transformations. ACM Trans. on Graphics. **1**, **3** (1982) 215–234
15. Yang, B., Böhme, J.: Reducing the computations of the singular value decomposition array given by Brent and Luk. SIAM J. Matrix Anal. Appl., **12**, **4** (1991) 713–725
16. Yoshimura, H., Nakanishi, T., Tamauchi, H.: A 50MHz CMOS geometrical mapping processor. IEEE Intl. Solid-State Circuits Conf. (1988) 162–164